UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/288,263	04/08/1999	HIROYUKI WAKI	NAK1-BG55	7236

21611 7590 06/12/2003

SNELL & WILMER LLP
1920 MAIN STREET
SUITE 1200
IRVINE, CA 92614-7230

EXAMINER
LAFORGIA, CHRISTIAN A

ART UNIT	PAPER NUMBER
2155	

DATE MAILED: 06/12/2003

16

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/288,263	WAKI ET AL.	
	Examiner	Art Unit	
	Christian La Forgia	2155	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 May 2003* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *29-31 and 39-53* is/are pending in the application.

4a) Of the above claim(s) *29,31 and 45-51* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *30,39-44,52 and 53* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *17 January 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. The amendment filed on 22 May 2003 is noted and made of record.

2. Claims 29-31 and 39-53 are presented for examination.

3. Claims 29, 31, and 45 through 51 have been cancelled as per Applicant's request.

Continued Examination Under 37 CFR 1.114

4. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 22 May 2003 has been entered.

Drawings

5. Applicant is reminded that the Patent and Trademark Office no longer makes drawing changes and that it is applicant's responsibility to ensure that the drawings are corrected in accordance with the instructions set forth in Paper No. 12, mailed on 20 February 2003.

Claim Rejections - 35 USC § 101

6. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

7. Claims 30, 39-44, 52 and 53 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

8. As per claims 30, 39-44, 52 and 53 merely claimed as a computer program representing a computer listing *per se*, that is, descriptions or expressions of such a program and that is, descriptive material *per se*, non-functional descriptive material, and is not statutory because it is

not a physical "thing" nor a statutory process, as there are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed aspects of the invention which permit the computer program's functionality to be realized. Since a computer program is merely a set of instructions capable of being executed by a computer, the program itself is not a process, without the computer-readable medium needed to realize the computer program's functionality. In contrast, a claimed computer-readable medium encoded with a computer program defines structural and functional interrelationships between the computer program and the medium which permit the computer program's functionality to be realized, and is thus statutory. **Warmerdam**, 33 F.3d at 1361, 31 USPQ2d at 1760. **In re Sarkar**, 588 F.2d 1330, 1333, 200 USPQ 132, 137 (CCPA 1978). See MPEP § 2106(IV)(B)(1)(a).

Claim Rejections - 35 USC § 112

9. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

10. Claims 30 and 54 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The applicant does not clearly define where or why the code is being transmitted.

11. Claim 43 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The applicant does not clearly define operation.

Claim Rejections - 35 USC § 102

12. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

13. Claims 30, 39, 53 and 54 are rejected under 35 U.S.C. 102(e) as being anticipated by United States Patent No. 6,170,083 to Adl-Tabatabai, (hereinafter Adl-Tabatabai).

14. As per claim 30, Adl-Tabatabai teaches a data structure that stores a virtual machine instruction sequence generated by compiler to be executed by a virtual machine, the data structure including:

15. a plurality of instruction blocks that constitute the virtual machine instruction sequence, the instruction blocks corresponding to separate basic blocks and formatted for transmission (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 38);

16. the instruction blocks including:

an identifier area for storing an identifier that specifies a start position of the instruction block (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9);

a non-branch instruction area for storing non-branch instructions belonging to a corresponding basic block (column 5, lines 36-55);

a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block (column 5, lines 36-55); and

17. each branch instruction stored in the branch instruction area designating a branch destination using an identifier stored in one of the identifier areas (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9). This is based on the definition of a basic block as offered by United States Patent No. 6,044,222 to Simons et al., in which a basic block is defined as

> A basic block is a contiguous sequence of instructions such that control flow can only enter at the first instruction and leave at the last instruction. Basic blocks have the characteristic that if one of the instructions is ever executed by the program, all the instructions in the basic block will be executed (assuming that no exceptions occur). A basic block typically starts at a branch label and is typically terminated by some sort of branch instruction. *Column 2, lines 15-23*

For a better graphical representation of a basic block please refer to United States Patent No. 5,923,883 to Tanaka et al., Figures 1, 2, 3, 4, 5, 6, 10, 12, 15, 16, 17, 18, 19, 20, 21, 23a, 23b, 24c, 26a, and 28b.

18. As per claim 39, Adl-Tabatabai teaches a data structure that stores a virtual machine instruction sequence generated by compiler to be executed by a virtual machine, the data structure including:

19. a plurality of instruction blocks that constitute the virtual machine instruction sequence, the instruction blocks corresponding to separate basic blocks and formatted for transmission (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 38);

20. the instruction blocks each including:

an identifier area for storing an identifier that specifies a start position of the instruction block (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9);

a non-branch instruction area for storing non-branch instructions belonging to the corresponding basic block (column 5, lines 36-55); and,

a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block (column 5, lines 36-55).

This is based on the definition of a basic block as offered by United States Patent No. 6,044,222 to Simons et al., in which a basic block is defined as

> A basic block is a contiguous sequence of instructions such that control flow can only enter at the first instruction and leave at the last instruction. Basic blocks have the characteristic that if one of the instructions is ever executed by the program, all the instructions in the basic block will be executed (assuming that no exceptions occur). A basic block typically starts at a branch label and is typically terminated by some sort of branch instruction. *Column 2, lines 15-23*

For a better graphical representation of a basic block please refer to United States Patent No. 5,923,883 to Tanaka et al., Figures 1, 2, 3, 4, 5, 6, 10, 12, 15, 16, 17, 18, 19, 20, 21, 23a, 23b, 24c, 26a, and 28b.

21. As per claim 52, Adl-Tabatabai teaches a virtual machine instruction sequence generated by compiler to be executed by a virtual machine, the improvement comprising:

22. a plurality of instruction blocks that constitute the virtual machine instruction sequence, the instruction blocks corresponding to basic blocks (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 38);

23. the instruction blocks each including:

an identifier area for storing an identifier that specifies a start position of the instruction block (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9);

a non-branch instruction area for storing non-branch instructions belonging to a corresponding basic block (column 5, lines 36-55);

a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block (column 5, lines 36-55); and,

each branch instruction stored in the branch instruction area designating a branch destination using an identifier stored in one of the identifier areas (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9),

wherein the division of the virtual machine instruction sequence into a plurality of separately identifiable instruction blocks having a single branch instruction area reduces the amount of branch destination processing that would otherwise be necessary with a single instruction sequence with branch instructions throughout (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9).

This is based on the definition of a basic block as offered by United States Patent No. 6,044,222 to Simons et al., in which a basic block is defined as

> A basic block is a contiguous sequence of instructions such that control flow can only enter at the first instruction and leave at the last instruction. Basic blocks have the characteristic that if one of the instructions is ever executed by the program, all the instructions in the basic block will be executed (assuming that no exceptions occur). A basic block typically starts at a branch label and is typically terminated by some sort of branch instruction. *Column 2, lines 15-23*

For a better graphical representation of a basic block please refer to United States Patent No. 5,923,883 to Tanaka et al., Figures 1, 2, 3, 4, 5, 6, 10, 12, 15, 16, 17, 18, 19, 20, 21, 23a, 23b, 24c, 26a, and 28b.

24. As per claim 54, Adl-Tabatabai teaches a virtual machine instruction sequence generated by a compiler to be executed by a virtual machine, the improvement comprising:

25. a plurality of instruction blocks that constitute the virtual machine instruction sequence, the instruction blocks corresponding to basic blocks (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 38);

26. the instruction blocks each including:

an identifier area for storing an identifier that specifies a start position of the instruction block (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9);

a non-branch instruction area for storing non-branch instructions belonging to a corresponding basic block (column 5, lines 36-55);

a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block (column 5, lines 36-55); and,

each branch instruction stored in the branch instruction area designating a branch destination using an identifier stored in one of the identifier areas (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9);

wherein the virtual machine instruction sequence is transmitted after being divided into a plurality of instruction blocks (Figures 1 [block 110], 2 [block 210], 3 [blocks 330, 331, 333, 340], 4, 6a, 6b, 7; column 2, line 65 to column 3, line 27; column 4, lines 26-43 ["after a Java class file has been divided into code regions, the Java Virtual Machine then begins to execute the code"]; column 5, line 28 to column 6, line 38).

This is based on the definition of a basic block as offered by United States Patent No. 6,044,222 to Simons et al., in which a basic block is defined as

> A basic block is a contiguous sequence of instructions such that control flow can only enter at the first instruction and leave at the last instruction. Basic blocks have the characteristic that if one of the instructions is ever executed by the program, all the instructions in the basic block will be executed (assuming that no exceptions occur). A

basic block typically starts at a branch label and is typically terminated by some sort of branch instruction. *Column 2, lines 15-23*

For a better graphical representation of a basic block please refer to United States Patent No. 5,923,883 to Tanaka et al., Figures 1, 2, 3, 4, 5, 6, 10, 12, 15, 16, 17, 18, 19, 20, 21, 23a, 23b, 24c, 26a, and 28b.

Claim Rejections - 35 USC § 103

27. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

28. Claims 40, 42, and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Adl-Tabatabai in view of United States Patent No. 6,301,652 to Prosser et al., (hereinafter Prosser).

29. As per claim 40, Adl-Tabatabai does not teach wherein the identifier of the instruction block is address related information in the virtual machine instruction sequence.

30. Prosser teaches wherein the identifier of the instruction block is address related information in the virtual machine instruction sequence (column 2, lines 47-60). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the identifiers [labels] of Adl-Tabatabai so that they included address related information much like that of Prosser. One would be motivated to modify the address labeling because it would make more efficient use of memory for execution by performing a cache aligning function. Therefore, one block would lead another block in terms of execution and storage in memory. See column 3, line 62 to column 4, line 2 of Prosser.

31. Regarding claim 42, Adl-Tabatabai a virtual machine instruction at the start position of the basic block being allocated to a specific address in the virtual machine instruction sequence (column 5, lines 36-55), and

32. a virtual machine instruction at other than the start position of the basic block being allocated to an address other than the specific address (column 5, lines 36-55). This is based on the definition of a basic block as offered by United States Patent No. 6,044,222 to Simons et al., in which a basic block is defined as

> A basic block is a contiguous sequence of instructions such that control flow can only enter at the first instruction and leave at the last instruction. Basic blocks have the characteristic that if one of the instructions is ever executed by the program, all the instructions in the basic block will be executed (assuming that no exceptions occur). A basic block typically starts at a branch label and is typically terminated by some sort of branch instruction. *Column 2, lines 15-23*

For a better graphical representation of a basic block please refer to United States Patent No. 5,923,883 to Tanaka et al., Figures 1, 2, 3, 4, 5, 6, 10, 12, 15, 16, 17, 18, 19, 20, 21, 23a, 23b, 24c, 26a, and 28b.

33. Adl-Tabatabai does not teach wherein whether each virtual machine instruction is positioned at a start position of the basic block is indicated by an address in the virtual machine instruction sequence to which the virtual machine instruction is allocated.

34. Prosser teaches wherein whether each virtual machine instruction is positioned at a start position of the basic block is indicated by an address in the virtual machine instruction sequence to which the virtual machine instruction is allocated (column 2, lines 47-60). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the identifiers [labels] of Adl-Tabatabai so that they included address related information much like that of Prosser. One would be motivated to modify the address labeling because it would make

more efficient use of memory for execution by performing a cache aligning function. Therefore, one block would lead another block in terms of execution and storage in memory. See column 3, line 62 to column 4, line 2 of Prosser.

35. Regarding claim 44, Adl-Tabatabai teaches the basic blocks (Figures 4 [block 430], 6a, 6b, 7; column 5, lines 28-55).

36. Adl-Tabatabai does not teach identification tags, each designates an address related information of the virtual machine instruction at a start position of the basic block; attachment of the tag indicating if the virtual machine instruction corresponding to the identification tag is positioned at the start position of the basic block.

37. Prosser teaches identification tags, each designates an address related information of the virtual machine instruction at a start position of the basic block; attachment of the tag indicating if the virtual machine instruction corresponding to the identification tag is positioned at the start position of the basic block (column 2, lines 47-60). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the identification tags [labels] of Adl-Tabatabai so that they included address related information much like that of Prosser. One would be motivated to modify the address labeling because it would make more efficient use of memory for execution by performing a cache aligning function. Therefore, one block would lead another block in terms of execution and storage in memory. See column 3, line 62 to column 4, line 2 of Prosser.

38. Claims 41 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Adl-Tabatabai in view of Prosser as applied to claim 40 above, and further in view of United States Patent No. 6,151,618 to Wahbe et al., (hereinafter Wahbe).

39. Regarding claim 41, Adl-Tabatabai and Prosser do not teach wherein the address related information is one of absolute address, relative address, and offset address.

40. Wahbe teaches wherein the address related information is one of absolute address, relative address, and offset address (Figures 4 [block 430], 6 [blocks 628, 632, 635], 8 [blocks 811]; column 12, lines 14-30). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the address information of Wahbe in the combined system of Adl-Tabatabai and Prosser. One would be motivated to include the address information because it would make more efficient use of memory for execution by performing memory optimization.

41. Regarding claim 43, Adl-Tabatabai teaches an operation specifying unit for specifying an operation to be executed by the virtual machine (Figure 3 [block 320], 7 [block 705]; column 4, lines 26-43).

42. Adl-Tabatabai does not teach an identifying unit for storing identification information which indicates if the virtual machine instruction is positioned at a start position of the basic block.

43. Wahbe teaches an identifying unit for storing identification information which indicates if the virtual machine instruction is positioned at a start position of the basic block (Figures 4 [block 430], 6 [blocks 625, 628], 8 [blocks 811]; column 12, lines 14-30). It would have been

obvious to one of ordinary skill in the art at the time the invention was made to include the identifying unit [stack pointer] of Wahbe in the combined system of Adl-Tabatabai and Prosser. One would be motivated to include the identifying unit because it would make more efficient use of memory for execution by performing memory optimization.

Conclusion

44. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Christian La Forgia whose telephone number is (703) 305-7704. The examiner can normally be reached on Monday thru Thursday 7-5.

45. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on (703) 305-9648. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7240 for regular communications and (703) 746-7239 for After Final communications.

46. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Christian La Forgia
Patent Examiner
Art Unit 2155

clf
June 4, 2003

AYAZ SHEIKH
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100